

CLAIMS

What is claimed is:

1. A method of managing visibility of GUI components in an application comprising:

5 initializing the application;

 invoking a visibility manager; and

 displaying a user interface of the application wherein display of the GUI components are determined by the visibility manager.

10 2. The method as recited in claim 1 wherein invoking the visibility manager comprises:

 reading one or more profiles;

 processing the one or more profiles;

 reading and processing a user configuration based on the one or more
15 profiles; and

 activating a particular profile of the one or more profiles.

3. The method as recited in claim 2 wherein invoking the visibility manager further comprises:

revising the user interface based on the particular profile of the one or more profiles; and

displaying the user interface.

- 5 4. The method as recited in claim 3 wherein initializing the application comprises:

starting the application;

building the user interface with all of the GUI components visible; and

10 calling the visibility manager after the particular profile of the one or more profiles is activated.

5. The method as recited in claim 2 wherein activating the particular profile of the one or more profiles comprises:

selecting an identification of a particular GUI component;

15 locating the identification in a mapping table;

checking a state of the particular GUI component;

comparing the state to the particular profile of the one or more profiles;

changing the state if not in agreement with the particular profile of the one or more profiles; and

repeating locating the identification, checking the state of the particular GUI component, comparing the state and changing the state for any remaining additional identifications of additional GUI components.

5 6. The method as recited in claim 5 wherein the state is visible or not visible.

7. The method as recited in claim 5 wherein the mapping table comprises a plurality of identifications of GUI components and a corresponding plurality of references to Java objects.

10

8. The method as recited in claim 1 further comprising:

invoking the visibility manager by:

reading one or more profiles,

processing the one or more profiles,

15 reading and processing a user configuration based on the one or more profiles,

activating a particular profile of the one or more profiles by:

selecting an identification of a particular GUI component,

locating the identification in a mapping table,

20 checking a state of the particular GUI component,

changing the state if not in agreement with the particular profile of the one or more profiles, and

repeating locating the identification, checking the state of the particular GUI component, comparing the state and changing the state for any remaining additional identifications of additional GUI components,

revising the user interface based on the particular profile of the one or more profiles, and

displaying the user interface;

initializing the application by:

starting the application,

building the user interface with all of the GUI components visible, and

calling the visibility manager after the particular profile of the one or more profiles is activated.

9. A system for managing visibility of GUI components in an application comprising:

a user interface module of the application, including the GUI components;

and

a visibility manager that determines which GUI components are visible.

10. The system as recited in claim 9, wherein

the visibility manager to determine which GUI components to display by,
and the visibility manager further to:

read one or more profiles upon initialization of the application;

5 process the one or more profiles;

read and process a user configuration based on the one or more profiles; and

activate a particular profile of the one or more profiles.

11. The system as recited in claim 10 wherein

10 the user interface module to:

build the user interface with all of the GUI components visible;

the visibility manager to:

call the visibility manager after the particular profile of the one or more
profiles is activated;

15 revise the user interface based on the particular profile of the one or more
profiles; and

the user interface module further to:

display the user interface.

12. The system as recited in claim 10 wherein the visibility manager to:

select an identification of a particular GUI component;

locate the identification in a mapping table;

check a state of the particular GUI component;

5 compare the state to the particular profile of the one or more profiles; and

change the state if not in agreement with the particular profile of the one or more profiles.

13. The system as recited in claim 12 wherein activating the particular profile of

10 the one or more profiles further comprises repeating locating the identification, checking the state of the particular GUI component, comparing the state and changing the state for any remaining additional identifications of additional GUI components.

15 14. The system as recited in claim 12 wherein the state is visible or not visible.

15. The system as recited in claim 12 wherein the mapping table comprises a plurality of identifications of GUI components and a corresponding plurality of references to Java objects.

20

16. The system as recited in claim 9 further wherein

the visibility manager to determine which GUI components to display by,
and the visibility manager to:

read one or more profiles upon initialization of the application,

5 process the one or more profiles,

read and process a user configuration based on the one or more
profiles,

activate a particular profile of the one or more profiles by:

select an identification of a particular GUI component,

10 locate the identification in a mapping table,

check a state of the particular GUI component,

compare the state to the particular profile of the one or more
profiles, and

15 change the state if not in agreement with the particular profile
of the one or more profiles,

start the application,

build the user interface with all of the GUI components visible,

call the visibility manager after the particular profile of the one or
more profiles is activated,

revise the user interface based on the particular profile of the one or more profiles, and

display the user interface.

5 17. A visibility manager data structure for use in managing visibility of GUI components in a user interface of an application comprising a mapping table, one or more profiles and a user configuration. /

18. The visibility manager data structure as recited in claim 17 wherein the
10 visibility manager data structure is to interact with a visibility manager, and the visibility manager is to determine which GUI components are visible in the application.

19. The visibility manager data structure as recited in claim 18, wherein
15 the visibility manager to determine which GUI components to display, and the visibility manager further to:

read the one or more profiles upon initialization of the application;

process the one or more profiles;

read and process the user configuration based on the one or more profiles;

20 and

activate a particular profile of the one or more profiles based upon the mapping table.

20. The visibility manager data structure as recited in claim 19 wherein

5 the visibility manager to further determine which GUI components are visible in the application, and the visibility manager further to:

revise the user interface based on the particular profile of the one or more profiles; and

display a user interface.

10

21. The visibility manager data structure as recited in claim 19 wherein the visibility manager to:

select an identification of a particular GUI component;

locate the identification in a mapping table;

15 check a state of a specific GUI component in the application;

compare the state to the particular profile of the one or more of profiles;

change the state if not in agreement with the particular profile of the one or more profiles; and

20 wherein the visibility manager to select the identification, locate the identification, check the state of the particular GUI component, compare the state

and change the state if not in agreement with the particular profile of the one or more profiles.

22. The visibility manager data structure as recited in claim 21 wherein the state
5 is visible or not visible.

23. The visibility manager data structure as recited in claim 17 wherein the mapping table comprises a plurality of identifications of GUI components and a corresponding plurality of references to Java objects.

10

24. The visibility manager data structure as recited in claim 18 wherein
the visibility manager to determine which GUI components to display, and
the visibility manager further to:

read the one or more profiles upon initialization of the application;

15 process the one or more profiles;

read and process the user configuration based on the one or more profiles;

activate a particular profile of the one or more profiles based upon the mapping table by:

20 select an identification of a particular GUI component,

locate the identification in a mapping table,
 check a state of a specific GUI component in the application,
 compare the state to the particular profile of the one or more of
 profiles, and

5 change the state if not in agreement with the particular profile
 of the one or more profiles;

revise the user interface based on the activated profile, and

display a user interface.

10 25. A medium embodying instructions, which, when executed by a processor,
 cause the processor to perform a method, the method comprising:

 initializing an application;

 invoking a visibility manager; and

 displaying a user interface of the application wherein display of the GUI

15 components are determined by the visibility manager.

26. The medium as recited in claim 25 wherein invoking the visibility manager
 comprises:

 reading one or more profiles;

20 processing the one or more profiles;

reading and processing a user configuration based on the one or more profiles;

activating a particular profile of the one or more profiles;

revising the user interface based on the particular profile of the one or more profiles; and

displaying the user interface.

27. The medium as recited in claim 26 wherein initializing the application comprises:

starting the application;

building the user interface with all of the GUI components visible; and

calling the visibility manager after the particular profile of the one or more profiles is activated.

28. The medium as recited in claim 26 wherein activating particular profile of the one or more profiles comprises:

selecting an identification of a particular GUI component;

locating the identification in a mapping table;

checking a state of the particular GUI component;

comparing the state to the particular profile of the one or more profiles; and

changing the state if not in agreement with the particular profile of the one or more profiles.

29. The medium as recited in claim 26 further comprising:

5 repeating locating the identification, checking the state of the particular GUI component, comparing the state and changing the state for any remaining additional identifications of additional GUI components.

30. The medium as recited in claim 28 wherein the state is visible or not visible.

10

31. The medium as recited in claim 28 wherein the mapping table comprises a plurality of identifications of GUI components and a corresponding plurality of references to Java objects.

15 32. The medium as recited in claim 25 further comprising:

invoking the visibility manager by:

reading one or more profiles,

processing the one or more profiles,

reading and processing a user configuration based on the one or more
20 profiles,

activating a particular profile of the one or more profiles by:

selecting an identification of a particular GUI component,

locating the identification in a mapping table,

checking a state of the particular GUI component,

5 changing the state if not in agreement with the particular profile
of the one or more profiles, and

repeating locating the identification, checking the state of the
particular GUI component, comparing the state and changing
the state for any remaining additional identifications of

10 additional GUI components,

revising the user interface based on the particular profile of the one or
more profiles, and

displaying the user interface;

initializing the application by:

15 starting the application,

building the user interface with all of the GUI components visible, and

calling the visibility manager after the profile is activated.

33. A system for managing visibility of GUI components in an application

20 comprising:

means for interfacing with a user, the means for interfacing including the GUI components; and

means for determining which GUI components are visible.

- 5 34. The system as recited in claim 33 wherein the means for determining includes:

means for reading one or more profiles upon initialization of the application;

means for processing the one or more profiles;

10 means for reading and processing a user configuration based on the one or more profiles; and

means for activating a particular profile of the one or more profiles.

35. The system as recited in claim 34 further comprising:

means for starting the application;

15 means for building the user interface with all of the GUI components visible;

means for determining which GUI components are visible after the particular profile of the one or more profiles is activated; and

means for displaying the user interface.